STATIONARITY RESULTS FOR GENERATING SET SEARCH FOR LINEARLY CONSTRAINED OPTIMIZATION

TAMARA G. KOLDA*, ROBERT MICHAEL LEWIS†, AND VIRGINIA TORCZON‡

Abstract. We derive new stationarity results for derivative-free, generating set search methods for linearly constrained optimization. We show that there is a measure of stationarity that is of the same order as the step length at an identifiable subset of the iterations. Thus, even in the absence of explicit gradient information, we still have information about stationarity. These results help both to unify the convergence analysis of several direct search algorithms and to clarify the fundamental geometrical ideas that underlie them. In addition, these results validate a practical stopping criterion for such algorithms, which numerical results confirm.

Key words. constrained optimization, direct search, generating set search, global convergence analysis, nonlinear programming, derivative-free methods, pattern search

AMS subject classifications. 90C56, 90C30, 65K05

1. Introduction.

We consider the convergence properties of a class of direct search methods for solving linearly constrained optimization problems:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & Ax \le b. \end{array} \tag{1.1}$$

Here $f : \mathbb{R}^n \to \mathbb{R}$ and A is an $m \times n$ matrix. We assume that f is continuously differentiable on Ω but that the gradient information is not computationally available; i.e., no procedure exists for computing the gradient and it cannot be approximated accurately. We use Ω to denote the feasible set:

$$\Omega = \{ \, x \;\mid\; Ax \le b \, \} \, .$$

We do not assume that the constraints are nondegenerate.

Even though direct search methods do not have explicit recourse to derivatives, there are direct search algorithms that have been shown to converge to Karush–Kuhn–Tucker (KKT) points of (1.1); e.g., see [27, 36, 21, 26, 31, 17]. Our goal in revisiting the convergence analysis of direct search algorithms for solving (1.1) is to explore the threads common to each of these analyses. Specifically, we present stationarity results in §6 that clarify the roles played by the choice of search directions, step lengths, and step acceptance criterion for the direct search algorithms considered. Additionally, since it is assumed that direct search methods do not have direct access to ∇f, we ask if it is possible to say anything about the quality of the solution returned by a direct search method when it is terminated; the answer for the class of direct search methods we consider here turns out to be "Yes."

---

*Computational Sciences and Mathematics Research Department, Sandia National Laboratories, Livermore, CA 94551-9217; `tgkolda@sandia.gov`. This work was supported by the Mathematical, Information, and Computational Sciences Program of the U.S. Department of Energy, under contract DE-AC04-94AL85000 with Sandia Corporation.

†Department of Mathematics, College of William & Mary, P.O. Box 8795, Williamsburg, Virginia, 23187-8795; `buckaroo@math.wm.edu`. This research was supported by the Computer Science Research Institute at Sandia National Laboratories and by the National Science Foundation under Grant DMS-0215444.

‡Department of Computer Science, College of William & Mary, P.O. Box 8795, Williamsburg, Virginia, 23187-8795; `va@cs.wm.edu`. This research was supported by the Computer Science Research Institute at Sandia National Laboratories.

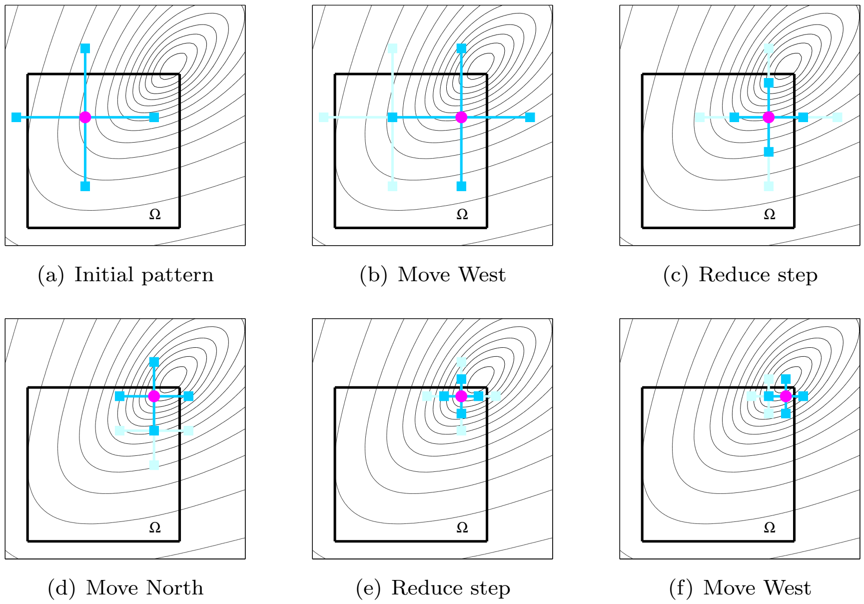


FIG. 1.1. *Compass search with exact penalization applied to the modified Broyden tridiagonal function with bound constraints.*

We refer to the direct search methods we examine (which include, for instance, pattern search methods [33]) as *generating set search* (GSS) [17]. To give some context for the discussion that follows, in Figure 1.1 we illustrate compass search, one particular instance of a GSS method. We apply compass search to a standard test problem (the two-dimensional modified Broyden tridiagonal function), with bounds on the variables. Level curves of the function are shown in the background; the feasible region is the box labeled Ω.

Compass search can be summarized as follows: Generate trial points (illustrated using squares) to the East, West, North, and South of the current iterate (illustrated using a circle). If one of the trial points is feasible *and* reduces the value of the function at the current iterate, the improved point becomes the new iterate and the iteration is deemed *successful*. Otherwise, none of the feasible trial points improves upon the value of the function at the current iterate, so we reduce the step by a factor of a half. In this case the iteration is deemed *unsuccessful*.

The kth iterate is denoted x_k. At each iteration, a set of search directions, $\mathcal{D}_k = \left\{ d_k^{(1)}, \ldots, d_k^{(p_k)} \right\}$, is generated. For compass search, we always choose $d_k^{(1)} = e_1$, $d_k^{(2)} = -e_1$, $d_k^{(3)} = e_2$ and $d_k^{(4)} = -e_2$, where the e_i's are the unit coordinate vectors. The trial points are then given by $\left\{ \, x_k + \Delta_k d_k^{(i)} \;\middle|\; i = 1, \ldots, 4 \, \right\}$. The scalar Δ_k is the step-length control parameter. The set of indices of all successful iterations is denoted by $\mathcal{S}$. The set of indices of all unsuccessful iterations is denoted by $\mathcal{U}$.

Note that in the example illustrated in Figure 1.1, as x_k approaches the constrained solution, compass search reduces the lengths of the steps taken by reducing Δ_k. Typically, compass search is terminated when the step length control parameter Δ_k falls below a certain tolerance, say Δ_{tol}. One of the goals of this paper is to show that this is a reasonable test for termination.

In the case of unconstrained minimization ($\Omega = \mathbb{R}^n$), it can be shown (under appropriate conditions) [11, 17] that the following relation holds:

$$\| \nabla f(x_k) \| = O(\Delta_k) \text{ for } k \in \mathcal{U}. \tag{1.2}$$

At the same time, one can also show that $\liminf_{k\to\infty} \Delta_k = 0$. As a consequence, one obtains the stationarity result $\liminf_{k\to\infty} \| \nabla f(x_k) \| = 0$.

In this paper we establish analogous results for (1.1). We consider the following measure of stationarity, which was introduced in [9]:

$$\chi(x) \equiv \max_{\substack{x+w\in\Omega \\ \| w \| \le 1}} -\nabla f(x)^T w.$$

As discussed in [10], $\chi(x)$ is a continuous function. Furthermore, $\chi(x) = 0$ for $x \in \Omega$ if and only if x is a KKT point of (1.1). In addition, if $\Omega = \mathbb{R}^n$, then $\chi(x) = \| \nabla f(x) \|$. We show in Theorem 6.4 that, under certain assumptions,

$$\chi(x_k) = O(\Delta_k) \text{ for } k \in \mathcal{U}. \tag{1.3}$$

The same arguments as in the unconstrained case show that $\liminf_{k\to\infty} \Delta_k = 0$. Thus, we once again obtain first-order stationarity. In addition, if $\Omega = \mathbb{R}^n$, then (1.3) reduces to (1.2), as one would wish.

By way of background, a different measure of stationarity was previously studied in [21] for the case of linear constraints. The quantity

$$q(x) \equiv P_\Omega\left(x - \nabla f(x)\right) - x, \tag{1.4}$$

where P_Ω denotes the projection onto Ω, is a continuous function of x with the property that $q(x) = 0$ for $x \in \Omega$ if and only if x is a KKT point. In [21, Proposition 7.1] it was shown that

$$\| q(x_k) \| = O(\sqrt{\Delta_k}) \text{ for } k \in \mathcal{U}. \tag{1.5}$$

If $\Omega = \mathbb{R}^n$, then $q(x) = -\nabla f(x)$; however, (1.5) does not reduce to (1.2) as one would wish (though in the bound constrained case, one can obtain a relationship of the form $\| q(x_k) \| = O(\Delta_k)$ [22]).

The algorithms discussed here are similar to other recently proposed direct search methods for bound and linearly constrained optimization [20, 21, 24, 26, 2, 31, 16]. We present a new criterion, melding ideas from [20, 21, 24, 26], for generating a sufficient set of search directions. The approach outlined in [21] and Algorithm 2 of [26] use one prescription for the set of search directions; Algorithm 1 in [26] uses another. Here we give yet a third, intermediate to both. What has become clear is that there is a variety of options for generating the set of search directions that guarantee asymptotic convergence to KKT points of (1.1).

The analysis in [2, 24, 26, 31] differs from our perspective in that the focus is on the nature of the limit points of the sequence of iterates (at least one limit point existing under mild and standard assumptions). In [26], the assumption that f is continuously differentiable then ensures that for their Algorithm 1, at least one limit point is a KKT point of (1.1), and for their Algorithm 2, any limit point is a KKT point of (1.1). The results in [2, 31], relax the smoothness requirements on f, but conclude only that *if* f is strictly differentiable [8] at a limit point, then that limit point is a KKT point of (1.1). However, without assuming more differentiability of

f there are no mechanisms within the algorithms proposed in [2, 31] to ensure that the sequence of iterates converges to a point that satisfies the KKT conditions. It is straightforward to construct examples for which the algorithms can fail to converge to KKT points [17].

Here we assume that f is continuously differentiable to ensure asymptotic convergence to a KKT point. However, rather than looking only at limit points, we examine what happens as the iterates progress toward a KKT point. This difference in perspective allows us to obtain a bound on $\chi(x_k)$ once the lengths of the steps are "small enough," where "small enough" is in a sense that we can quantify. This difference in perspective has practical import for those interested in implementing GSS methods for solving (1.1). For instance, the bound on χ implied by (1.3) and obtained in (6.6) supports our assertion that Δ_k can be used to assess progress toward a solution. Furthermore, the bound in (6.6) illuminates what algorithmic parameters can—and should—be monitored to ensure the effectiveness of an implementation [19]. Finally, our analysis also yields an estimate (Theorem 6.3) that makes it possible to use the direct search algorithms we discuss in connection with the augmented Lagrangian approach presented in [9] to handle problems with both linear and nonlinear constraints.

The paper is organized as follows. In §2, we describe how to choose search directions for GSS methods applied to problems with linear constraints. As we saw in Figure 1.1, GSS algorithms may generate trial points that are infeasible, so in §3 we describe how feasibility is maintained. In §4 we discuss the globalization strategies that ensure $\liminf_{k\to\infty} \Delta_k \to 0$. Formal statements of GSS algorithms for solving linearly constrained problems are given in §5. We present two general algorithms. The first (Algorithm 5.1) uses a *sufficient decrease* condition, similar to that used in [24, 26]). The second (Algorithm 5.2) uses a *simple decrease* condition as in [20, 21]. Results showing that these algorithms converge to a KKT point of (1.1) are derived in §6. In §7 we discuss what the analysis reveals about using Δ_k to test for convergence and demonstrate its effectiveness on two test problems. In §8, we summarize the results and their importance. Appendix A contains a discussion of $\chi(x)$ and its use as a measure of stationarity. Appendix B contains geometric results on cones and polyhedra.

2. Search directions. Since GSS methods assume that $\nabla f(x)$ is not available, they cannot directly identify descent directions. As a first consideration, therefore, GSS methods must include enough search directions in the set $\mathcal{D}_k$ to guarantee that at least one of them is a descent direction, no matter the direction of steepest descent. In addition, there must be a way to ensure that at least one of the search directions in $\mathcal{D}_k$ allows for a sufficiently long step while remaining inside the feasible region. In this context, "sufficiently long" is relative to the amount of possible local improvement that can be made. This demand requires a careful choice of search directions when the iterate x_k is near the boundary of the feasible region, so GSS methods for solving problem (1.1) require mechanisms for choosing directions that capture the geometry of the feasible region near x_k.

To describe what is required of the sets of search directions for GSS methods for linearly constrained problems, we start in §2.1 by reviewing some basic concepts regarding finitely-generated cones. Then, in §2.2, we show how to use the constraints given in (1.1) to define cones that mirror the geometry of the boundary of the polyhedron Ω near the current iterate x_k. Finally, in §2.3, we show how to dynamically construct a set $\mathcal{D}_k$ that satisfies our requirement that there exists at least one direction along which it is possible to take a step of sufficient length while remaining inside Ω.

2.1. Cones and generators. A set K is a cone if for any $x \in K$, $\alpha x \in K$ for all $\alpha \geq 0$. The polar of a cone K, denoted K°, is defined by

$$K^\circ = \{\ v \ \mid \ w^T v \leq 0 \text{ for all } w \in K\ \}.$$

The polar of a cone is itself a cone.

Given a convex cone K and any vector v, there is a unique closest point of K to v, called the *projection* of v onto K, which we denote by v_K. Given a vector v and a convex cone K, any vector v can be written as $v = v_K + v_{K^\circ}$ and $v_K^T v_{K^\circ} = 0$ [28, 32, 34, 37].

A set of vectors $\mathcal{G}$ *generates* K if K is the set of all nonnegative linear combinations of elements of $\mathcal{G}$. A cone K is called finitely generated if it can be generated by a finite set of vectors. If K is finitely generated by $\mathcal{G}$, then we define

$$\kappa(\mathcal{G}) = \min_{\substack{v \in \mathbb{R}^n \\ v_K \neq 0}} \max_{d \in \mathcal{G}} \frac{v^T d}{\|\, v_K \,\|\, \|\, d \,\|}. \tag{2.1}$$

This is a generalization of the quantity given in [17, (3.10)], where it was assumed that $K = \mathbb{R}^n$. Proposition 2.1 says that $\kappa(\mathcal{G}) > 0$; see Corollary 10.4 in [21].

PROPOSITION 2.1. *Let K be a convex cone in $\mathbb{R}^n$ generated by the finite set $\mathcal{G}$. Then $\kappa(\mathcal{G}) > 0$.*

2.2. Tangent and normal cones. Recalling the definition of the linearly constrained problem (1.1), let a_i^T be the ith row of the constraint matrix A, and let

$$\mathcal{C}_i = \{\ y \ \mid \ a_i^T y = b_i\ \}$$

denote the set where the ith constraint is binding. The set $I(x)$ of binding constraints at x is $I(x) = \{\ i \ \mid \ x \in \mathcal{C}_i\ \}$. The *normal cone* at a point x, denoted by $N(x)$, is the cone generated by the binding constraints, i.e., the cone generated by the set $\{\ a_i \ \mid \ i \in I(x)\ \} \cup \{0\}$. The presence of $\{0\}$ means that $N(x) = \{0\}$ if there are no binding constraints. The *tangent cone*, denoted by $T(x)$, is the polar of the normal cone. Further discussion of the tangent and polar cones in the context of optimization can be found, for instance, in [7, 30, 32].

In our case, we are not only interested in the binding constraints, but also in the nearby constraints. Given $x \in \Omega$, we define the set of ε-*binding constraints* to be

$$I(x, \varepsilon) = \{\ i \ \mid \ \operatorname{dist}(x, \mathcal{C}_i) \leq \varepsilon\ \}. \tag{2.2}$$

The vectors a_i for $i \in I(x, \varepsilon)$ are the outward-pointing normals to the faces of the boundary of Ω within distance ε of x. Examples are shown in Figure 2.1 for three choices of $x \in \Omega$.

Given $x \in \Omega$, we define the ε-*normal cone* $N(x, \varepsilon)$ to be the cone generated by the set $\{\ a_i \ \mid \ i \in I(x, \varepsilon)\ \} \cup \{0\}$. Its corresponding polar cone is the ε-*tangent cone* $T(x, \varepsilon)$. Observe that if $\varepsilon = 0$, these are just the standard normal and tangent cones; that is, $N(x, 0) = N(x)$ and $T(x, 0) = T(x)$.

Examples of ε-normal and ε-tangent cones are illustrated in Figure 2.2. The cone $T(x, \varepsilon)$ approximates the polyhedron Ω near x. (More precisely, $x + T(x, \varepsilon)$ approximates the feasible region near x.) Note that if $I(x, \varepsilon) = \emptyset$, then $N(x, \varepsilon) = \{0\}$

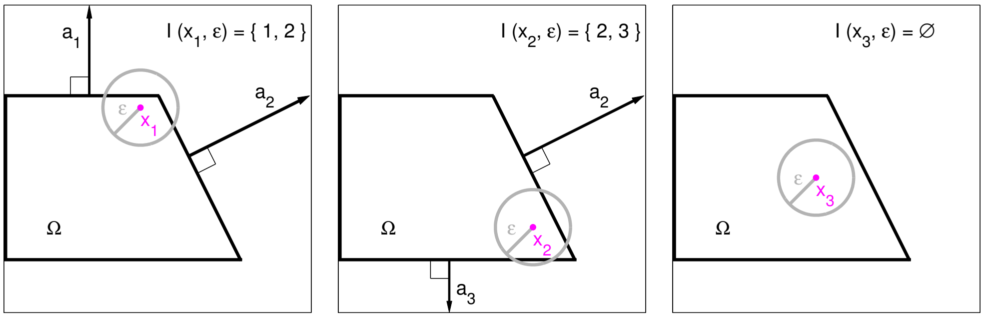


FIG. 2.1. *The outward-pointing normals a_i for the index set $I(x_1, \varepsilon) = \{1, 2\}$ and a_i for the index set $I(x_2, \varepsilon) = \{2, 3\}$. Since the distance from x_3 to $\partial\Omega$ is greater than ε, $I(x_3, \varepsilon) = \emptyset$.*

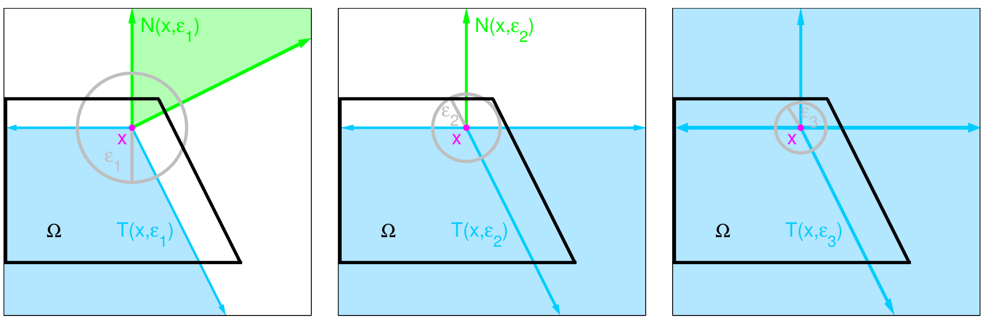


FIG. 2.2. *The cones $N(x, \varepsilon)$ and $T(x, \varepsilon)$ for the values ε_1, ε_2, and ε_3. Note that for this example, as ε varies from ε_1 to 0, there are only the three distinct pairs of cones illustrated ($N(x, \varepsilon_3) = \{0\}$).*

and $T(x, \varepsilon) = \mathbb{R}^n$; in other words, if the boundary is more than distance ε away then the problem looks unconstrained in the ε-neighborhood of x, as can be seen in the third example in Figure 2.2.

If $T(x, \varepsilon) \neq \{0\}$, then one can proceed from x along all directions in $T(x, \varepsilon)$ for a distance of at least ε, and remain inside the feasible region, as illustrated in Figure 2.2 and formalized in Proposition 2.2.

PROPOSITION 2.2. *If $x \in \Omega$, and $v \in T(x, \varepsilon)$ satisfies $\| v \| \leq \varepsilon$, then $x + v \in \Omega$.*

Proof. Suppose not; i.e., $v \in T(x, \varepsilon)$ and $\| v \| \leq \varepsilon$, but $x + v \notin \Omega$. Then there exists i such that $a_i^T(x + v) > b_i$. Using the fact that $x \in \Omega$, so $a_i^T x \leq b_i$, we have

$$a_i^T v > b_i - a_i^T x \geq 0. \tag{2.3}$$

Define

$$t = \frac{b_i - a_i^T x}{a_i^T v}.$$

Note that $t < 1$ by (2.3). Let $y = x + tv$. Then $a_i^T y = b_i$ and $\| x - y \| = \| tv \| < \varepsilon$. Thus, $i \in I(x, \varepsilon)$ and $a_i \in N(x, \varepsilon)$. Since, by hypothesis, $v \in T(x, \varepsilon)$, we must have $a_i^T v \leq 0$. However, this contradicts (2.3). ☐

All possible cones $N(x, \varepsilon)$ and $T(x, \varepsilon)$ can be generated using a finite number of vectors, as formalized in the following proposition.

PROPOSITION 2.3. *There exists a finite set* $\mathbf{A}$ *such that* $\mathbf{A}$ *contains generators for every set* $N(x,\varepsilon)$ *with* $x \in \Omega$ *and* $\varepsilon \geq 0$. *As a consequence, there also exists a finite set* $\mathbf{G}$ *such that* $\mathbf{G}$ *contains generators for every set* $T(x,\varepsilon)$ *with* $x \in \Omega$ *and* $\varepsilon \geq 0$.

Proof. Let $\mathbf{A} = \{a_1, \ldots, a_m\}$. Each $N(x,\varepsilon)$ is generated by at most m vectors (one per ε-binding constraint) from the set $\mathbf{A}$.

Furthermore, since $N(x,\varepsilon)$ is finitely generated, its polar $T(x,\varepsilon)$ is also finitely generated [32, 34]. Next, the set of all possible ε-normal cones is finite. More specifically, the cardinality of the set $\{ N(x,\varepsilon) \mid x \in \Omega, \ \varepsilon > 0 \}$ is bounded by 2^m where m is the number of linear constraints. Thus, there are only finitely many sets $T(x,\varepsilon)$, each of which is finitely generated. Choose a finite set of generators for each possible $T(x,\varepsilon)$; then the union $\mathbf{G}$ of these generators is finite. ∎

The definition of $\mathbf{A}$ ensures that for every $x \in \Omega$ and $\varepsilon \geq 0$ there exists $\mathcal{A} \subseteq \mathbf{A}$ such that $\mathcal{A}$ generates $N(x,\varepsilon)$. Furthermore, for every $x \in \Omega$ and $\varepsilon \geq 0$, there exists a constant $\nu_{\min} > 0$ such that the following holds:

$$\kappa(\mathcal{A}) \geq \nu_{\min}. \tag{2.4}$$

The set $\mathbf{A}$ is immediate. The set $\mathbf{G}$ is not; nevertheless, such a set is possible to construct. Returning to the example in Figure 2.2, in Figure 2.3 we show generators of $T(x,\varepsilon)$ for x, which is fixed, with three different values of ε. In Figure 2.3, $d^{(1)}$ and $d^{(2)}$ generate $T(x,\varepsilon_1)$; $d^{(1)}$, $d^{(2)}$, and $d^{(3)}$ generate $T(x,\varepsilon_2)$; and $d^{(1)}$, $d^{(2)}$, $d^{(3)}$, and $d^{(4)}$ generate $T(x,\varepsilon_3)$. For this simple example it is a straightforward exercise to verify that for any $x \in \Omega$ and any $\varepsilon > 0$, generators for every $T(x,\varepsilon)$ are contained in the finite set

$$\mathbf{G} = \{d^{(1)}, d^{(2}, d^{(3)}, d^{(4)}, d^{(5)}, d^{(6)}\}, \tag{2.5}$$

where $d^{(5)} = -d^{(1)}$ and $d^{(6)} = -d^{(4)}$.

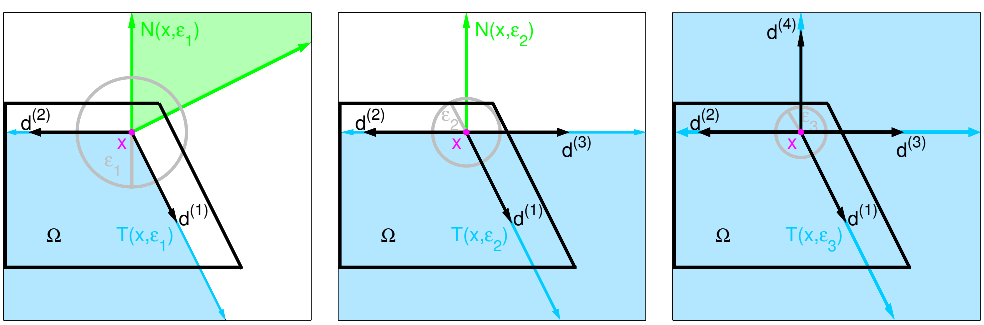


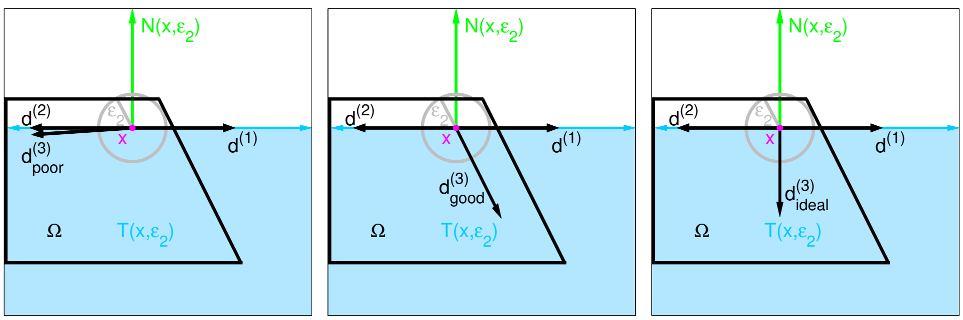
FIG. 2.3. *Generators for the cones* $T(x,\varepsilon)$ *for the values* ε_1, ε_2, *and* ε_3.

2.3. Choosing the search directions. We can now state the conditions we place on the search directions for GSS for linearly constrained optimization. At each iteration k we require that a subset of the search directions generates $T(x_k,\varepsilon_k)$. The quantity $\varepsilon_k > 0$ is the threshold for deciding which constraints are considered nearby at iteration k. The particular value of ε_k, discussed in §5, involves the maximum step length at iteration k. The requirement that $\mathcal{D}_k$ must contain vectors which generate the cone $T(x_k,\varepsilon_k)$ ensures that $\mathcal{D}_k$ contains at least one search direction along which a sufficiently long feasible step can be taken.

To make the condition precise, recall that the set of search directions at iteration k is $\mathcal{D}_k$. We partition $\mathcal{D}_k$ so that $\mathcal{D}_k = \mathcal{G}_k \cup \mathcal{H}_k$. The set $\mathcal{G}_k$ contains the directions that are critical to ensure convergence and so is the focus of our analysis. The (possibly empty) set $\mathcal{H}_k$ contains extra directions that may improve the effectiveness of the search. The set $\mathcal{G}_k$ must satisfy Condition 1.

CONDITION 1. There exists a constant $\kappa_{\min} > 0$, independent of k, such that for all k the following holds. For every k, there exists a $\mathcal{G} \subseteq \mathcal{G}_k$ such that $\mathcal{G}$ generates $T(x_k, \varepsilon_k)$ and, furthermore, $\kappa(\mathcal{G}) \geq \kappa_{\min}$.

The lower bound $\kappa_{\min}$ is needed to avoid the situation shown in Figure 2.4(a). The "ideal" choice, using a minimal number of vectors while ensuring that $\kappa(\mathcal{G})$ is as large as possible, would be $d^{(3)}_{\text{ideal}}$, as shown in Figure 2.4(c).



(a) Poor choice (b) Acceptable choice (c) "Ideal" choice

FIG. 2.4. *Condition 1 is needed to avoid a poor choice when there is freedom in choosing* $\mathcal{G}$.

There are many ways to satisfy the requirement $\kappa(\mathcal{G}) \geq \kappa_{\min}$. The simplest is to draw $\mathcal{G}_k$ from the finite set $\mathbf{G}$ that exists by Proposition 2.3.

The analysis presented here uses the geometric ideas in [21], which are purely in terms of the intrinsic geometry of the feasible region rather than any particular algebraic representation. *We do* not *assume that the constraints are nondegenerate.* That said, as a practical matter, identifying the generators for $T(x, \varepsilon)$ can require appreciable work when the constraints are degenerate, or nearly so. Fortunately, this is a problem that has been well-studied in computational geometry, and both solutions [29, 5, 6] and software [3, 4, 12] exist. Thus, as discussed in more detail in [19], it is possible to implement the linearly constrained GSS algorithms analyzed here, even in the presence of either degeneracy or linear equality constraints.

Condition 1 dictates how, at a minimum, the search directions should be directionally distributed; the next condition (Condition 2) imposes upper and lower bounds on the lengths of those directions.

CONDITION 2. There exist $\beta_{\min} > 0$ and $\beta_{\max} > 0$, independent of k, such that for all k the following holds.

$$\beta_{\min} \leq \|d\| \leq \beta_{\max} \qquad \text{for all } d \in \mathcal{G}_k.$$

This second condition on the choice of $\mathcal{G}_k$ is straightforward to satisfy, say, by normalizing all search directions so that $\beta_{\min} = \beta_{\max} = 1$. However, there may be situations where it makes sense to allow the directions in $\mathcal{G}_k$ to accommodate scaling information. This poses no difficulties for the analysis, so long as there are lower and upper bounds, independent of k, on the norm of each $d \in \mathcal{G}_k$.

3. Choosing the step lengths. Given a set of search directions, the length of the step along each direction is dictated by the *step-length control parameter*, denoted Δ_k. In the unconstrained case, the set of trial points at iteration k would be

$$\left\{ \, x_k + \Delta_k^{(i)} d_k^{(i)} \ \mid \ i = 1, \ldots, p_k \, \right\},$$

where

$$\mathcal{D}_k = \left\{ d_k^{(1)}, d_k^{(2)}, \ldots, d_k^{(p_k)} \right\}.$$

In the constrained case, however, some of those trial points may be infeasible. Thus, the trial points are instead defined by

$$\left\{ \, x_k + \tilde{\Delta}_k^{(i)} d_k^{(i)} \ \mid \ i = 1, \ldots, p_k \, \right\},$$

where

$$\tilde{\Delta}_k^{(i)} \in [0, \Delta_k]$$

is chosen so that $x_k + \tilde{\Delta}_k^{(i)} d_k^{(i)} \in \Omega$. (The actual step lengths depend both on $\tilde{\Delta}_k$ and on $\| \, d \, \|$, but recall that $\| \, d \, \|$ is bounded by Condition 2.) The main requirement on choosing $\tilde{\Delta}_k^{(i)}$ is that a full step is used if possible, as formally stated in the following condition.

CONDITION 3. If $x_k + \Delta_k d_k^{(i)} \in \Omega$, then $\tilde{\Delta}_k^{(i)} = \Delta_k$.

The simplest formula for choosing $\tilde{\Delta}_k^{(i)} \in [0, \Delta_k]$ that satisfies Condition 3 is

$$\tilde{\Delta}_k^{(i)} = \begin{cases} \Delta_k & \text{if } x_k + \Delta_k d_k^{(i)} \in \Omega \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

This corresponds to exact penalization (see the discussion in Section 8.1 of [17]) since the effect of (3.1) is to reject (by setting $\tilde{\Delta}_k^{(i)} = 0$) any step $\Delta_k d_k^{(i)}$ that would generate an infeasible trial point. Since the definition of (1.1) assumes that the constraints are explicit (i.e., $Ax \le b$ is known), verifying the feasibility of a trial point is straightforward.

Exact penalization is illustrated in Figure 3.1. Returning to our example from §2, we have $\mathcal{G}_k = \left\{ d_k^{(1)}, d_k^{(2)}, d_k^{(3)}, d_k^{(4)} \right\}$, where $\mathcal{G}_k \subset \mathbf{G}$, with the set $\mathbf{G}$ from (2.5) and $d_k^{(1)} \equiv d^{(1)}$, $d_k^{(2)} \equiv d^{(2)}$, $d_k^{(3)} \equiv d^{(3)}$, and $d_k^{(4)} \equiv d^{(5)}$. Since $x_k + \Delta_k d_k^{(2)}$ and $x_k + \Delta_k d_k^{(4)}$ are feasible, as shown on the left, Condition 3 requires $\tilde{\Delta}_k^{(2)} = \Delta_k$ and $\tilde{\Delta}_k^{(4)} = \Delta_k$, as shown on the right. On the other hand, since $x_k + \Delta_k d_k^{(1)}$ and $x_k + \Delta_k d_k^{(3)}$ are infeasible, as shown on the left, (3.1) sets $\tilde{\Delta}_k^{(1)} = 0$ and $\tilde{\Delta}_k^{(3)} = 0$, as shown on the right.

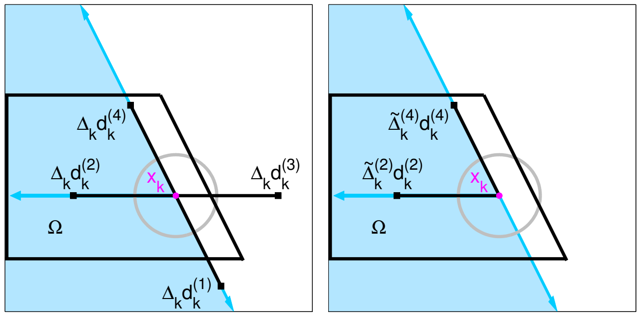


FIG. 3.1. *The step-length control parameter Δ_k may lead to infeasible trial points. The effect of exact penalization is that infeasible points simply are not considered as candidates to replace x_k.*

More sophisticated strategies can be employed for choosing $\tilde{\Delta}_k^{(i)}$ when $x_k + \Delta_k d_k^{(i)}$ is infeasible. The rationale for choosing, whenever possible, a step length $\tilde{\Delta}_k^{(i)} \in (0, \Delta_k]$ to ensure that the trial point $x_k + \tilde{\Delta}_k^{(i)} d_k^{(i)}$ is feasible is simple enough: it seems unsatisfactory to have gone to some pains to ensure at least one feasible descent direction only to reject the resulting trial point due to the value of Δ_k. Since our alternatives for choosing $\tilde{\Delta}_k^{(i)}$ depend on the globalization strategy, we defer the discussion of further examples to §4.

4. Globalization. Globalization of GSS refers to the conditions that are enforced to ensure that

$$\liminf_{k \to \infty} \Delta_k = 0. \tag{4.1}$$

These conditions impact the decision of whether or not to accept a trial point as the next iterate and how to update Δ_k.

Globalization strategies for GSS are discussed in detail in [17]. Here we review those features that are relevant to our analysis of algorithms for the linearly constrained case.

In any GSS algorithm, x_k is always the best feasible point discovered thus far; i.e., $f(x_k) \le f(x_j)$ for all $j \le k$; however, different conditions are imposed on *how much better* a trial point must be to be accepted as the next iterate.

In general, for an iteration to be considered *successful*, we require that there exist $d_k \in \mathcal{D}_k$ and an $\tilde{\Delta}_k \in [0, \Delta_k]$ such that

$$x_k + \tilde{\Delta}_k d_k \in \Omega \quad \text{and} \quad f(x_k + \tilde{\Delta}_k d_k) < f(x_k) - \rho(\Delta_k). \tag{4.2}$$

The function $\rho(\cdot)$ is called the *forcing function* and must satisfy Condition 4:

CONDITION 4. (General requirements on the forcing function)
1. The function $\rho(\cdot)$ is a nonnegative continuous function on $[0, +\infty)$.
2. The function $\rho(\cdot)$ is $o(t)$ as $t \downarrow 0$; i.e., $\lim_{t \downarrow 0} \rho(t) \,/\, t = 0$.
3. The function $\rho(\cdot)$ is nondecreasing; i.e., $\rho(t_1) \le \rho(t_2)$ if $t_1 \le t_2$.

Both $\rho(\Delta) \equiv 0$ and $\rho(\Delta) = \alpha \Delta^p$, where α is some positive scalar and $p > 1$, satisfy Condition 4. The first choice can only be used with globalization via a rational lattice, which is discussed in §4.2. The second choice can be used with globalization via a sufficient decrease condition, which is discussed in §4.1.

In the case of a successful iteration (i.e., one that satisfies (4.2)), the next iterate is defined by

$$x_{k+1} = x_k + \tilde{\Delta}_k d_k \text{ for } k \in \mathcal{S}.$$

(Recall that the set of indices of all successful iterations is denoted by $\mathcal{S}$.) In addition, Δ_k is updated according to

$$\Delta_{k+1} = \phi_k \Delta_k, \ \phi_k \geq 1 \text{ for } k \in \mathcal{S}.$$

The parameter ϕ_k is called the *expansion parameter*.

If the kth iteration is unsuccessful, it then must be the case that for each $d \in \mathcal{G}_k$,

$$\text{either } x_k + \Delta_k d \notin \Omega \quad \text{or} \quad f(x_k + \Delta_k d) \geq f(x_k) - \rho(\Delta_k).$$

In this case, the best point is unchanged, i.e.,

$$x_{k+1} = x_k \text{ for } k \in \mathcal{U},$$

(recall that the set of indices of all unsuccessful iterations is denoted by $\mathcal{U}$) and the step-length control parameter is reduced:

$$\Delta_{k+1} = \theta_k \Delta_k, \ \theta_k \in (0,1), \ \text{for } k \in \mathcal{U}.$$

The parameter θ_k is called the *contraction parameter*.

There are intimate connections between choosing the ϕ_k or θ_k in the update for Δ_k and guaranteeing that (4.1) holds. Further requirements depend on the particular choice of globalization strategy, and so are given in §4.1 and §4.2.

4.1. Globalization via a sufficient decrease condition. In the context of gradient-based nonlinear programming algorithms, the enforcement of a sufficient decrease condition on the step is well-established (e.g., [1, 13, 35], or see the general discussion in Section 2.2 of [17]). Enforcing the classic sufficient decrease condition ties the choice of the step-length control parameter to the expected decrease, as estimated by the initial rate of decrease $-\nabla f(x_k)^T d_k$. In the context of GSS methods, the underlying assumption is that the value of $\nabla f(x_k)$ is unavailable—which means that the classic sufficient decrease condition cannot be enforced. However, in [14] an alternative that uses the step-control parameter, rather than $\nabla f(x_k)$, was introduced and analyzed in the context of linesearch methods for unconstrained minimization. This basic concept then was extended to GSS methods, for both the unconstrained and the constrained case, in [23, 24, 25, 26]. We restate the essential features here.

Within the context of GSS methods for linearly constrained optimization, a sufficient decrease globalization strategy requires the following of the forcing function $\rho(\cdot)$ and the choice of the contraction parameter θ_k.

Condition 5. (The forcing function for sufficient decrease)
The forcing function $\rho(\cdot)$ is such that $\rho(t) > 0$ for $t > 0$.

Condition 6. (Decreasing Δ_k for sufficient decrease)
A constant $\theta_{\max} < 1$ exists such that $\theta_k \leq \theta_{\max}$ for all k.

Full details are discussed in Section 3.7.1 of [17], but we include a few salient observations here. First, the requirements of Condition 5 are easily satisfied by choosing, say, $\rho(\Delta) = 10^{-4}\Delta^2$, while the requirements of Condition 6 are easily satisfied by choosing, say, $\theta_k = \frac{1}{2}$ for all $k = 0, 1, \ldots$. Second, the upper bound on the contraction factor θ_k ensures a predictable fraction of reduction on Δ_k after an unsuccessful iteration, which we require to ensure (4.1) holds.

If a sufficient decrease condition is being employed, then we can use an alternate strategy to (3.1) for choosing $\tilde{\Delta}_k^{(i)}$, as follows:

$$\begin{array}{ll} \text{maximize} & \Delta \\ \text{subject to} & 0 \le \Delta \le \Delta_k, \\ & x_k + \Delta\, d_k^{(i)} \in \Omega. \end{array} \tag{4.3}$$

This option is illustrated in Figure 4.1.

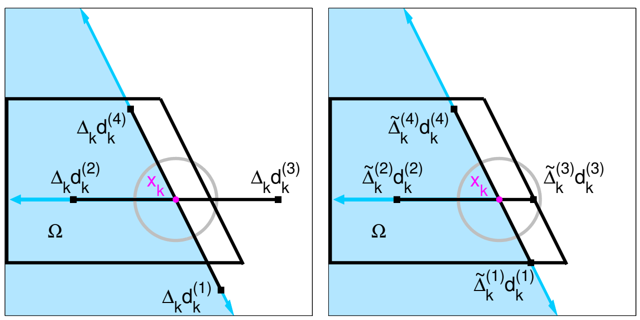


FIG. 4.1. *Observe in the second illustration that applying (4.3) pulls infeasible trial points to the boundary of Ω.*

4.2. Globalization via a rational lattice. Classical direct search methods rely on simple, as opposed to sufficient, decrease when accepting a step. In other words, it is enough for the step $\tilde{\Delta}_k^{(i)} d_k^{(i)}$ to satisfy $f(x_k + \tilde{\Delta}_k^{(i)} d_k^{(i)}) < f(x_k)$. The trade-off is that when the condition for accepting a step is relaxed, further restrictions are required on the types of steps that are allowed. These restrictions are detailed in Conditions 7, 8, and 9.

CONDITION 7. (Choosing the directions for the rational lattice)
1. The set $\mathbf{G} = \{d^{(1)}, \ldots, d^{(p)}\}$ is required to be a finite set of search directions.
2. Every vector $d \in \mathbf{G}$ is required to be of the form $d = Bc$, where $B \in \mathbb{R}^{n\times n}$ is a nonsingular matrix, and $c \in \mathbb{Q}^n$, where $\mathbb{Q}$ is the set of rational numbers.
3. All generators are chosen from $\mathbf{G}$; i.e., $\mathcal{G}_k \subseteq \mathbf{G}$ for all k.
4. All extra directions are integer combinations of the elements of $\mathbf{G}$;. i.e., $\mathcal{H}_k \subset \left\{ \sum_{i=0}^{p} \xi^{(i)} d^{(i)} \mid \xi^{(i)} \in \{0, 1, 2, \ldots\} \right\}$.

CONDITION 8. (Increasing or decreasing Δ_k for the rational lattice)
1. The scalar Λ is a fixed integer strictly greater than 1.
2. For all $k \in \mathcal{S}$, ϕ_k is of the form $\phi_k = \Lambda^{\ell_k}$ where $\ell_k \in \{0, 1, 2 \ldots\}$.
3. For all $k \in \mathcal{U}$, θ_k is of the form $\theta_k = \Lambda^{m_k}$ where $m_k \in \{-1, -2, \ldots\}$.

CONDITION 9. (Choosing the steps for the rational lattice)
All steps $\tilde{\Delta}_k \in [0, \Delta_k]$ satisfy either $\tilde{\Delta}_k = 0$ or $\tilde{\Delta}_k = \Lambda^m \Delta_k > \Delta_{\text{tol}}$, where $m \in \mathbb{Z}$, $m \leq 0$ and Δ_{tol} is a value independent of k satisfying $\Delta_0 > \Delta_{\text{tol}} > 0$.

While the list of requirements in Conditions 7, 8, and 9 looks onerous, in fact all can be satisfied in a straightforward fashion. A full discussion of the reasons for these requirements can be found in Section 3.7.2 of [17]; here we limit ourselves to a few observations.

First, a critical consequence of Conditions 7, 8, and 9 is that when all three are enforced, all iterates lie on a rational lattice, which plays a crucial role in ensuring (4.1) when only simple decrease is enforced.

Note also that the set $\mathbf{G}$ is a conceptual construct that describes a finite set of all admissible search directions. For instance, as specified in (2.5) for the example in Figure 2.3, $\mathbf{G}$ can be chosen to containing generators for all possible cones $T(x, \varepsilon)$, for all $x \in \Omega$ and all $\varepsilon \geq 0$. In this way the third requirement in Condition 7, necessary for globalization, is satisfied. The finiteness of $\mathbf{G}$ also means that Condition 1 and Condition 2 are satisfied automatically. Furthermore, it is not necessary to construct the set $\mathbf{G}$ of all potential search directions in advance. Finally, a standard assumption in the context of simple decrease [21] is that the linear constraints are rational, i.e., $A \in \mathbb{Q}^{m \times n}$; see [21, Section 8] for more details.

Now, consider the requirements on the scaling of the step given in Condition 8. The usual choice of Λ is 2. In the unconstrained case, ϕ_k typically is chosen from the set $\{1, 2\}$ so that $\ell_k \in \{0, 1\}$ for all k, satisfying the requirement placed on ϕ_k in Condition 8. Usually θ_k is chosen to be $\frac{1}{2}$ so that $m_k = -1$ for all k, satisfying the requirement placed on θ_k in Condition 8. Note that the fact Λ^{-1} is the largest possible choice of θ_k obviates the need to explicitly bound θ_k, as was required in Condition 6 for sufficient decrease.

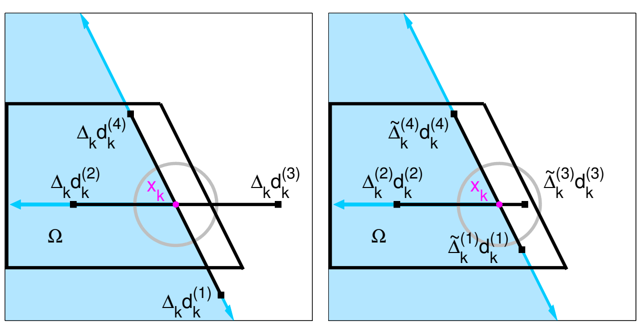


FIG. 4.2. *Observe in the illustration on the right that starting with $\tilde{\Delta}_k^{(i)} = \Delta_k$ and applying (4.4) has the effect of systematically decreasing $\tilde{\Delta}_k^{(i)}$ (in this example, by $\frac{1}{2}$) until $x_k + \tilde{\Delta}_k^{(i)} d^{(i)}$ is feasible (assuming Δ_{tol} is small enough).*

Consider Condition 9 says that is is possible to choose a partial step along a given direction so long as the trial point remains on the rational lattice. So, as illustrated in Figure 4.2, along direction $d^{(1)}$, $\tilde{\Delta}_k^{(1)} = 0.5\Delta_k$ yields the feasible trial step $\tilde{\Delta}_k^{(1)} d^{(1)}$ while along direction $d^{(3)}$, $\tilde{\Delta}_k^{(3)} = 0.25\Delta_k$ yields the feasible trial step $\tilde{\Delta}_k^{(3)} d^{(3)}$. These choices for $\tilde{\Delta}_k^{(1)}$ and $\tilde{\Delta}_k^{(3)}$ correspond to solving the following problem:

$$
\begin{array}{lll}
 & \text{maximize} & \Delta \\
(4.4) & \text{subject to} & \Delta = 0 \text{ or } \Delta = \Lambda^m \Delta_k, \text{ with } \Delta_{\text{tol}} < \Lambda^m \Delta_k, \; m \in \{-1, -2, \ldots\}, \\
 & & x_k + \Delta\, d_k^{(i)} \in \Omega.
\end{array}
$$

If, for instance, x_k is on the boundary of the feasible region, then it may be necessary to set $\tilde{\Delta}_k^{(i)}$ to zero if there is no feasible step from x_k along direction $d^{(i)}$. Otherwise, the goal is to take the longest possible step that keeps the trial point feasible while remaining on the rational lattice that underlies the search.

5. GSS algorithms for linearly constrained problems. We now formally state two GSS algorithms for solving problem (1.1). The primary requirements on the methods are that they satisfy Conditions 1, 2, 3, and 4. The differences in the two versions depend on the type of globalization that is used.

For both algorithms, the search directions must include generators for $T(x_k, \varepsilon_k)$. In [21] and Algorithm 2 of [26], and earlier in [27] (in a slightly restricted form), this is accomplished by requiring the search directions to contain generators for $T(x_k, \varepsilon)$ for all ε in the interval $[0, \varepsilon_{\max}]$, with $\varepsilon_{\max} > 0$. This condition turns out to include more directions than necessary. Algorithm 1 in [26] allows for a smaller set of search directions: the set of search directions must exactly generate $T(x_k, \varepsilon_k)$—and *only* $T(x_k, \varepsilon_k)$. For Algorithm 1 in [26] ε_k is simply a parameter decreased at unsuccessful iterations, as opposed to the particular ε_k we choose here. We can relax the condition that the set of search directions exactly generate only $T(x_k, \varepsilon_k)$ because we enforce a slightly more stringent sufficient decrease condition.

Our requirement that the search directions include generators for $T(x_k, \varepsilon_k)$ makes sense geometrically. The maximum length step we can try at iteration k is $\varepsilon_k = \Delta_k \beta_{\max}$, so none of the steps we try will encounter any part of the boundary more than a distance ε_k from x_k.

We note a technical difference between the presentation of the algorithms in Figures 5.1 and 5.2 and what is assumed for the analysis in §6. In practice, GSS algorithms terminate when the step-length control parameter Δ_k falls below a given threshold $\Delta_{\text{tol}} > 0$. Because this is important to any implementation, we have included it in the statement of the algorithm. In Theorems 6.3, 6.4, and 6.5, however, we assume that the iterations continue ad infinitum (i.e., in the context of the analysis, the reader should assume $\Delta_{\text{tol}} = 0$).

5.1. GSS using a sufficient decrease condition. A linearly constrained GSS algorithm based on a sufficient decrease globalization strategy is presented in Figure 5.1. Using a sufficient decrease globalization strategy, as outlined in §4.1, requires that we enforce two particular conditions. Condition 5 ensures that $\rho(\Delta_k) = 0$ only when $\Delta_k = 0$. Condition 6 ensures that there is sufficient reduction on Δ_k at unsuccessful iterations.

The only assumption on f necessary to show that some subsequence of the Δ_k converges to zero is that f be bounded below in the feasible region.

THEOREM 5.1 (Theorem 3.4 of [17]). *Suppose f is bounded below on Ω. Then*

ALGORITHM 5.1 (LINEARLY CONSTRAINED GSS USING A SUFFICIENT DECREASE GLOBALIZATION STRATEGY)

INITIALIZATION.

Let $x_0 \in \Omega$ be the initial guess.

Let $\Delta_{\text{tol}} > 0$ be the tolerance used to test for convergence.

Let $\Delta_0 > \Delta_{\text{tol}}$ be the initial value of the step-length control parameter.

Let $\varepsilon_{\max} > \beta_{\max}\Delta_{\text{tol}}$ be the maximum distance used to identify nearby constraints ($\varepsilon_{\max} = +\infty$ is permissible).

Let $\rho(\cdot)$ be a forcing function satisfying Conditions 4 and 5.

ALGORITHM. For each iteration $k = 0, 1, 2, \ldots$

STEP 1. Let $\varepsilon_k = \min\{\varepsilon_{\max}, \beta_{\max}\Delta_k\}$. Choose a set of search directions $\mathcal{D}_k = \mathcal{G}_k \cup \mathcal{H}_k$ satisfying Conditions 1 and 2.

STEP 2. If there exists $d_k \in \mathcal{D}_k$ and a corresponding $\tilde{\Delta}_k \in [0, \Delta_k]$ satisfying Condition 3 such that $x_k + \tilde{\Delta}_k d_k \in \Omega$ and

$$f(x_k + \tilde{\Delta}_k d_k) < f(x_k) - \rho(\mathbf{\Delta_k}),$$

then:

- Set $x_{k+1} = x_k + \tilde{\Delta}_k d_k$.
- Set $\Delta_{k+1} = \phi_k \Delta_k$ for any choice of $\phi_k \geq 1$.

STEP 3. Otherwise, for every $d \in \mathcal{G}_k$, either $x_k + \Delta_k d \notin \Omega$ or

$$f(x_k + \Delta_k d) \geq f(x_k) - \rho(\mathbf{\Delta_k}).$$

In this case:

- Set $x_{k+1} = x_k$ (no change).
- Set $\Delta_{k+1} = \theta_k \Delta_k$ for some choice $\theta_k \in (0, 1)$ satisfying **Condition 6**.

If $\Delta_{k+1} < \Delta_{\text{tol}}$, then terminate.

FIG. 5.1. *Linearly constrained GSS using a sufficient decrease globalization strategy.*

for a linearly constrained GSS method using a sufficient decrease globalization strategy satisfying Conditions 4, 5, and 6 (as outlined in Figure 5.1), $\liminf_{k\to\infty} \Delta_k = 0$.

5.2. GSS using a rational lattice. A linearly constrained GSS algorithm based on a rational lattice globalization strategy is presented in Figure 5.2. The choice $\rho(\cdot) \equiv 0$ is standard for the rational lattice globalization strategy, which means only simple decrease, i.e., $f(x_k + \tilde{\Delta}_k d_k) < f(x_k)$, is required. We note, however, that a sufficient decrease condition may be employed in conjunction with a rational lattice globalization strategy; see [17]. The choice $\rho(\cdot) \equiv 0$ also means that Condition 4

Algorithm 5.2 (Linearly constrained generating set search using a rational lattice globalization strategy)

Initialization.

Let $x_0 \in \Omega$ be the initial guess.

Let $\Delta_{\text{tol}} > 0$ be the tolerance used to test for convergence.

Let $\Delta_0 > \Delta_{\text{tol}}$ be the initial value of the step-length control parameter.

Let $\varepsilon_{\max} > \beta_{\max}\Delta_{\text{tol}}$ be the maximum distance used to identify nearby constraints ($\varepsilon_{\max} = +\infty$ is permissible).

Algorithm. For each iteration $k = 0, 1, 2, \ldots$

Step 1. Let $\varepsilon_k = \min\{\varepsilon_{\max}, \beta_{\max}\Delta_k\}$. Choose a set of search directions $\mathcal{D}_k = \mathcal{G}_k \cup \mathcal{H}_k$ satisfying Conditions 1, 2, **and 7**.

Step 2. If there exists $d_k \in \mathcal{D}_k$ and a corresponding $\tilde{\Delta}_k \in [0, \Delta_k]$ satisfying Conditions 3 **and 9** such that $x_k + \tilde{\Delta}_k d_k \in \Omega$ and

$$f(x_k + \tilde{\Delta}_k d_k) < f(x_k),$$

then:

– Set $x_{k+1} = x_k + \tilde{\Delta}_k d_k$.

– Set $\Delta_{k+1} = \phi_k \Delta_k$ for a choice of $\phi_k \geq 1$ **satisfying Condition 8**.

Step 3. Otherwise, for every $d \in \mathcal{G}_k$, either $x_k + \Delta_k d \notin \Omega$ or

$$f(x_k + \Delta_k d) \geq f(x_k).$$

In this case:

– Set $x_{k+1} = x_k$ (no change).

– Set $\Delta_{k+1} = \theta_k \Delta_k$ for some choice $\theta_k \in (0, 1)$ satisfying **Condition 8**.

If $\Delta_{k+1} < \Delta_{\text{tol}}$, then terminate.

Fig. 5.2. *Linearly constrained GSS using a rational lattice globalization strategy.*

is satisfied automatically. The trade-off for using simple decrease is that additional conditions must be imposed on the choice of admissible $\mathcal{D}_k$ (Condition 7), ϕ_k and θ_k (Condition 8), and $\tilde{\Delta}_k$ (Condition 9).

Using a rational lattice globalization strategy, to show that some subsequence of the step length control parameters goes to zero, the only assumption we place on f is that the set $\mathcal{F} = \{ x \in \Omega \mid f(x) \leq f(x_0) \}$ is bounded.

Theorem 5.2 (Theorem 3.8 of [17]). *Assume that $\mathcal{F} = \{ x \in \Omega \mid f(x) \leq f(x_0) \}$ is bounded. Then for a linearly constrained GSS methods using a lattice globalization strategy satisfying Conditions 4, 7, 8, and 9 (as outlined in Figure 5.2),*

$\liminf_{k\to\infty} \Delta_k = 0$.

6. Stationarity results. At *unsuccessful* iterations of the linearly constrained GSS methods outlined in Figures 5.1 and 5.2, we can bound the measure of stationarity $\chi(x_k)$ in terms of Δ_k. To do so, we make the following assumptions.

ASSUMPTION 6.1. *The set $\mathcal{F} = \{ \, x \in \Omega \mid f(x) \le f(x_0) \, \}$ is bounded.*

ASSUMPTION 6.2. *The gradient of f is Lipschitz continuous with constant M on $\mathcal{F}$.*

If both Assumption 6.1 and Assumption 6.2 hold, then there exists $\gamma > 0$ such that for all $x \in \mathcal{F}$,

$$\| \nabla f(x) \| < \gamma. \tag{6.1}$$

We then have the following results for the algorithms in Figures 5.1 and 5.2.

THEOREM 6.3. *Suppose that Assumption 6.2 holds. Consider the linearly constrained GSS algorithms given in Figures 5.1 and 5.2, both of which satisfy Conditions 1, 2, and 3. If k is an unsuccessful iteration and ε_k satisfies $\varepsilon_k = \beta_{\max}\Delta_k$, then*

$$\| [-\nabla f(x_k)]_{T(x_k,\varepsilon_k)} \| \le \frac{1}{\kappa_{\min}} \left(M\Delta_k\beta_{\max} + \frac{\rho(\Delta_k)}{\Delta_k\beta_{\min}} \right). \tag{6.2}$$

Here, $\kappa_{\min}$ is from Condition 1, M is from Assumption 6.2, and $\beta_{\max}$ and $\beta_{\min}$ are from Condition 2.

Proof. Clearly, we need only consider the case when $[-\nabla f(x_k)]_{T(x_k,\varepsilon_k)} \neq 0$. Condition 1 guarantees a set $\mathcal{G} \subseteq \mathcal{G}_k$ that generates $T(x_k,\varepsilon_k)$. By (2.1) (with $K = T(x_k,\varepsilon_k)$ and $v = -\nabla f(x_k)$) there exists some $\hat{d} \in \mathcal{G}$ such that

$$\kappa(\mathcal{G}) \, \| [-\nabla f(x_k)]_{T(x_k,\varepsilon_k)} \| \, \| \hat{d} \| \le -\nabla f(x_k)^T \hat{d}. \tag{6.3}$$

Condition 3 and the fact that iteration k is unsuccessful tell us that

$$f(x_k + \Delta_k d) \ge f(x_k) - \rho(\Delta_k) \quad \text{for all } d \in \mathcal{G}_k \text{ for which } x_k + \Delta_k d \in \Omega.$$

Condition 2 ensures that for all $d \in \mathcal{G}$, $\| \Delta_k d \| \le \Delta_k\beta_{\max}$ and, by assumption, $\Delta_k\beta_{\max} = \varepsilon_k$, so we have $\| \Delta_k d \| \le \varepsilon_k$ for all $d \in \mathcal{G}$. Proposition 2.2 then assures us that $x_k + \Delta_k d \in \Omega$ for all $d \in \mathcal{G}$. Thus,

$$f(x_k + \Delta_k d) - f(x_k) + \rho(\Delta_k) \ge 0 \quad \text{for all } d \in \mathcal{G}. \tag{6.4}$$

Meanwhile, since the gradient of f is assumed to be continuous (Assumption 6.2), we can apply the mean value theorem to obtain, for some $\alpha_k \in (0,1)$,

$$f(x_k + \Delta_k d) - f(x_k) = \Delta_k \nabla f(x_k + \alpha_k\Delta_k d)^T d \quad \text{for all } d \in \mathcal{G}.$$

Putting this together with (6.4),

$$0 \le \Delta_k \nabla f(x_k + \alpha_k\Delta_k d)^T d + \rho(\Delta_k) \qquad \text{for all } d \in \mathcal{G}.$$

Dividing through by Δ_k and subtracting $\nabla f(x_k)^T d$ from both sides yields

$$-\nabla f(x_k)^T d \le \left(\nabla f(x_k + \alpha_k \Delta_k d) - \nabla f(x_k)\right)^T d + \rho(\Delta_k)/\Delta_k \qquad \text{for all } d \in \mathcal{G}.$$

Since $\nabla f(x)$ is Lipschitz continuous (Assumption 6.2) and $0 < \alpha_k < 1$, we obtain

$$-\nabla f(x_k)^T d \le M \Delta_k \| d \|^2 + \rho(\Delta_k)/\Delta_k \qquad \text{for all } d \in \mathcal{G}. \tag{6.5}$$

Since (6.5) holds for all $d \in \mathcal{G}$, (6.3) tells us that for some $\hat{d} \in \mathcal{G}$,

$$\kappa(\mathcal{G}) \, \| \, [-\nabla f(x_k)]_{T(x_k,\varepsilon_k)} \, \| \le M \Delta_k \| \, \hat{d} \, \| + \frac{\rho(\Delta_k)}{\Delta_k \, \| \, \hat{d} \, \|}.$$

Using the bounds on $\| \, \hat{d} \, \|$ in Condition 2,

$$\| \, [-\nabla f(x_k)]_{T(x_k,\varepsilon_k)} \, \| \le \frac{1}{\kappa(\mathcal{G})} \left(M \Delta_k \beta_{\max} + \frac{\rho(\Delta_k)}{\Delta_k \beta_{\min}} \right).$$

The theorem then follows from the fact that $\kappa(\mathcal{G}) > \kappa_{\min}$ (Condition 1). $\square$

Theorem 6.4 relates the measure of stationarity $\chi(x_k)$ to the step length control parameter Δ_k.

THEOREM 6.4. *Suppose that Assumptions 6.1 and 6.2 hold. Consider the linearly constrained GSS algorithms given in Figures 5.1 and 5.2, both of which satisfy Conditions 1, 2, and 3. If k is an unsuccessful iteration and $\varepsilon_k = \beta_{\max}\Delta_k$, then*

$$\chi(x_k) \le \left(\frac{M}{\kappa_{\min}} + \frac{\gamma}{\nu_{\min}} \right) \Delta_k \, \beta_{\max} + \frac{1}{\kappa_{\min} \, \beta_{\min}} \frac{\rho(\Delta_k)}{\Delta_k}. \tag{6.6}$$

Here, $\kappa_{\min}$ is from Condition 1, $\nu_{\min}$ is from (2.4), M is from Assumption 6.2, γ is from (6.1), and $\beta_{\max}$ and $\beta_{\min}$ are from Condition 2.

Proof. Since $\varepsilon_k = \Delta_k \beta_{\max}$, Proposition B.2 tells us that

$$\chi(x_k) \le \| \, [-\nabla f(x_k)]_{T(x_k,\varepsilon_k)} \, \| + \frac{\Delta_k \beta_{\max}}{\nu_{\min}} \, \| \, [-\nabla f(x_k)]_{N(x_k,\varepsilon_k)} \, \|.$$

Furthermore, the bound on $\| \, [-\nabla f(x_k)]_{T(x_k,\varepsilon_k)} \, \|$ from Theorem 6.3 holds. The projection onto convex sets is contractive, so $\| \, [-\nabla f(x_k)]_{N(x_k,\varepsilon_k)} \, \| \le \| \, \nabla f(x_k) \, \|$. Under Assumptions 6.1 and 6.2, (6.1) holds, so $\| \, [-\nabla f(x_k)]_{N(x_k,\varepsilon_k)} \, \| \le \gamma$. The result follows. $\square$

If we choose either $\rho(\Delta) \equiv 0$ or $\rho(\Delta) = \alpha \Delta^p$ with $\alpha > 0$ and $p \ge 2$, then we obtain an estimate of the form $\chi(x_k) = O(\Delta_k)$.

The constants M, γ, and $\nu_{\min}$ in (6.6) are properties of the problem (1.1). The remaining quantities—the bounds on the lengths of the search directions $\beta_{\min}$ and $\beta_{\max}$, as well as $\kappa_{\min}$—are under the control of the algorithm. The value of $\kappa_{\min}$ can be increased by adding search directions to $\mathcal{G} \subseteq \mathcal{G}_k$.

Before we continue, we observe that the Lipschitz assumption (Assumption 6.2) can be relaxed. A similar bound can be obtained assuming only continuous differentiability of f. Let ω denote the following modulus of continuity of $\nabla f(x)$: given $x \in \Omega$ and $r > 0$,

$$\omega(x, r) = \max \left\{ \, \| \, \nabla f(y) - \nabla f(x) \, \| \;\; \middle| \;\; y \in \Omega, \; \| \, y - x \, \| \le r \, \right\}.$$

Then the proof of Theorem 6.4 yields the bound

$$\chi(x_k) \leq \frac{1}{\kappa_{\min}}\,\omega(x_k, \Delta_k\beta_{\max}) + \frac{\gamma}{\nu_{\min}}\,\Delta_k\,\beta_{\max} + \frac{1}{\kappa_{\min}\,\beta_{\min}}\,\frac{\rho(\Delta_k)}{\Delta_k}.$$

Returning to Theorem 6.4, if we recall from Theorems 5.1 and 5.2 that the step-length control parameter Δ_k is manipulated explicitly by GSS methods in a way that ensures $\liminf_{k\to\infty}\Delta_k = 0$, then an immediate corollary is the following first-order convergence result.

THEOREM 6.5. *Suppose that Assumptions 6.1 and 6.2 hold. Then for the linearly constrained GSS algorithm in Figure 5.1, which satisfies Conditions 1, 2, 3, 4, 5, and 6, as well as for the linearly constrained GSS algorithm in Figure 5.2, which satisfies Conditions 1, 2, 3, 4, 7, 8, and 9, we have*

$$\liminf_{k\to 0}\,\chi(x_k) = 0.$$

7. Using Δ_k to terminate GSS methods after unsuccessful iterations. We now present some numerical illustrations of the practical implications of Theorem 6.4. We show that Δ_k can be used as a reasonable measure of stationarity when implementing GSS methods to solve (1.1). The results in §6 serve as a justification for terminating the search when $\Delta_k < \Delta_{\mathrm{tol}}$.

To demonstrate that Δ_k is a reasonable measure of stationarity, we show the following results from experiments using an implementation of a GSS method for solving (1.1) (a thorough discussion of the implementation, as well as further numerical results, can be found in [19]).

The first test problem is the following quadratic program (QP) for $n = 8$:

$$\begin{array}{ll} \text{minimize} & f(x) = \sum_{j=1}^{n} j^2 x_j^2 \\ \text{subject to} & 0 \leq x \leq 1 \\ & \sum_{j=1}^{n} x_j \geq 1, \end{array} \tag{7.1}$$

where x_j is the jth component of the vector x. The last constraint is binding at the solution. The second test problem is posed on a pyramid in $\mathbb{R}^3$:

$$\begin{array}{ll} \text{minimize} & f(x) = \sum_{j=1}^{3}[(4-j)^2(x_j - c_j)^2 - x_j] \\ \text{subject to} & x_3 \geq 0 \\ & x_1 + x_2 + x_3 \leq 1 \\ & x_1 - x_2 + x_3 \leq 1 \\ & -x_1 + x_2 + x_3 \leq 1 \\ & -x_1 - x_2 + x_3 \leq 1, \end{array} \tag{7.2}$$

with $c = (0.01, 0.01, 0.98)^T$. Again, x_j and c_j are the jth components of the vectors x and j, respectively. The solution is at c, which is near the apex of the pyramid. The algorithm actually visits the apex, which is a degenerate vertex insofar as there are four constraints in three variables that meet there.

These problems were solved using the forcing functions $\rho(\Delta) = 10^{-4}\Delta^2$ and $\rho(\Delta) \equiv 0$; the algorithm behaved exactly the same for both choices. The search directions included generators for the cones $T(x_k, \varepsilon)$ for all ε in the interval $[0, \varepsilon_{\max}]$, where $\varepsilon_{\max} = 0.2$. All search directions were normalized, so $\beta_{\min} = \beta_{\max} = 1$. For these choices, Theorem 6.4 says that $\chi(x_k) = O(\Delta_k)$ at unsuccessful iterations when $\Delta_k \leq \varepsilon_{\max}$.

After any unsuccessful iteration, we reduced Δ_k by half. Before proceeding to the next iteration, we recorded the value of Δ_k and computed the value of $\chi(x_k)$. These values are reported in Table 7.1 for iterations at which $\Delta_k \beta_{\max} \leq \varepsilon_{\max}$. As the results make clear, there are no real surprises.

(a) The QP in (7.1)

Δ_k	$\chi(x_k)$
0.0 50000000000	0 .708966400000
0.0 25000000000	0 .702676200000
0.0 12500000000	0 .702676200000
0.00 6250000000	0 .175954400000
0.00 3125000000	0 .175954400000
0.00 1562500000	0.0 50660390000
0.000 781250000	0.0 50660390000
0.000 390625000	0.00 6906371000
0.000 195312500	0.00 6906371000
0.0000 97656250	0.00 4441546000
0.0000 48828125	0.00 4441546000
0.0000 24414063	0.00 1171873000
0.0000 12207031	0.000 157063000
0.00000 6103516	0.000 157063000
0.00000 3051758	0.0000 10256150
0.00000 1525879	0.0000 10256150
0.000000 762939	0.00000 8368884

(b) The QP in (7.2)

Δ_k	$\chi(x_k)$
0.0 50000000000	0.0 77888810000
0.0 25000000000	0.0 77888810000
0.0 12500000000	0.0 31473920000
0.00 6250000000	0.00 9044181000
0.00 3125000000	0.00 9044181000
0.00 1562500000	0.00 3492924000
0.000 781250000	0.000 279221500
0.000 390625000	0.000 279221500
0.000 195312500	0.000 279221500
0.0000 97656250	0.000 279221500
0.0000 48828125	0.0000 93105050
0.0000 24414063	0.0000 24547840
0.0000 12207031	0.00000 9678778
0.00000 6103516	0.00000 2978927
0.00000 3051758	0.0000000 11121
0.00000 1525879	0.0000000 11121
0.000000 762939	0.0000000 11121

Table 7.1

`GSS` *runs showing decrease in* Δ_k *versus decrease in* $\chi(x_k)$ *at unsuccessful iterations.*

The point of the results report in Table 7.1 is not to demand close scrutiny of each entry but rather to demonstrate the trend in the quantities measured. We clearly see the linear relationship between Δ_k and $\chi(x_k)$ that Theorem 6.4 tells us to expect. These results are consistent with findings for the unconstrained case [11] as well as with a long-standing recommendation for using Δ_k as a stopping criterion for direct search methods (see [15]).

One practical benefit of using Δ_k as a measure of stationarity is that it is already present in GSS algorithms; no additional computation is required. Another good reason for using Δ_k as a measure of stationarity is that it is largely insusceptible to numerical error. Since GSS methods often are recommended when the evaluations of f are subject to numerical "noise," the fact that Δ_k will not be affected by numerical noise in the computed values of $f(x_k)$ suggests that Δ_k provides a particularly suitable stopping criterion.

We close with the observation that the effectiveness of Δ_k as a measure of stationarity clearly depends on the value of the constants in the bound in (6.6). For instance, if f is highly nonlinear, so that the Lipschitz constant M is large, then using Δ_k to estimate $\chi(x_k)$ might be misleading. While GSS methods cannot control M and γ, which depend on the problem (1.1), a careful implementation of GSS methods for solving (1.1) *can* control all the remaining constants in (6.6). Thus a careful implementation can ensure that Δ_k *is* a useful measure of stationarity except when f is highly nonlinear (i.e., M is large with respect to $\| \nabla f \|$).

8. Conclusions. The results we have presented are useful in several ways. First, they clarify the relationship between the geometry of the feasible region near an iterate x_k and the search directions needed by GSS. Theorem 6.3 and Theorem 6.4 bring out many of the elements common to the approaches described in [20, 21] and [25, 26]. Although the globalization approaches to ensure $\liminf_{k\to\infty} \Delta_k = 0$ differ, both classes of algorithms can use the same analysis to show that

$$\chi(x_k) = O(\Delta_k),$$

as described in connection with Theorem 6.4. This result does not depend on the method of globalization; instead, it depends mainly on choosing search directions that correctly reflect the nearby boundary.

Theorem 6.3 will allow the use of linearly constrained GSS methods in the augmented Lagrangian framework given in [9]. That approach proceeds by successive approximate minimization of the augmented Lagrangian. The stopping criterion in the subproblems involves the norm of the projection onto $T(x_k, \omega_k)$ of the negative gradient of the augmented Lagrangian, for a parameter $\omega_k \downarrow 0$. In the derivative-free setting the gradient is unavailable. However, Theorem 6.3 enables us to use Δ_k as an alternative measure of stationarity in the subproblems. Details will appear in [18].

An interesting consequence of these results is that we have theoretical support for terminating the algorithm when Δ_k falls below some tolerance. Theorem 6.4 shows that at the subsequence of unsuccessful iterations, we have $\chi(x_k) = O(\Delta_k)$ as $\Delta_k \to 0$. This is illustrated numerically in §7. At the same time, Theorem 6.4 also suggests that this stopping criterion may be unsuitable if the objective is highly nonlinear, making clear the need for direct search methods, like all optimization algorithms, to account for scaling.

In summary, we have provided a new take on direct search methods for linearly constrained problems. Our results illuminate how the properties of the problem (such as the nonlinearity of f and the boundary defined by A) and the parameter choices for the algorithm (such as the scaling and distribution of the search directions) will affect the progress of GSS towards a KKT point of problem (1.1). We have given a theoretical basis for a practical stopping criterion and provided numerical examples as justification.

Acknowledgments. We thank Margaret Wright, the associate editor who handled this paper, along with two anonymous referees, for the many useful comments that led to a significant improvement in the presentation.

Appendix A. Criticality measure for first-order constrained stationarity. Here we discuss $\chi(x)$ and $\| q(x) \|$ in more detail. Because these measures are not novel, we have relegated their discussion to an appendix.

As discussed in the introduction, for $x \in \Omega$ progress toward a KKT point of (1.1) can be measured by:

$$\chi(x) \equiv \max_{\substack{x+w\in\Omega \\ \| w \| \le 1}} -\nabla f(x)^T w. \tag{A.1}$$

This measure was originally proposed in [9] and is discussed at length in Section 12.1.4 of [10], where the following properties are noted:

1. $\chi(x)$ is continuous.

2. $\chi(x) \geq 0$.
3. $\chi(x) = 0$ if and only if x is a KKT point for (1.1).

Showing that $\chi(x_k) \to 0$ as $k \to \infty$ constitutes a global first-order stationarity result.

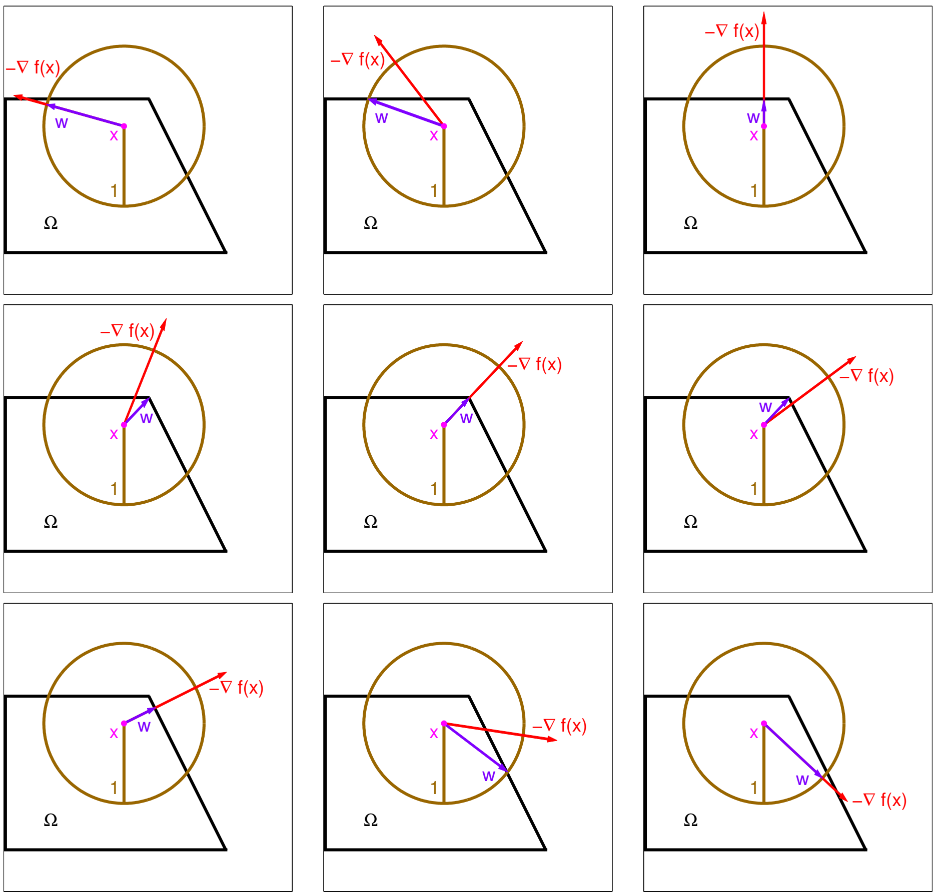


FIG. A.1. *How the w in (A.1) varies with $-\nabla f(x)$ when $x - \nabla f(x) \notin \Omega$.*

To help better understand this measure, the w's that define $\chi(x)$ in (A.1) are illustrated in Figure A.1 for several choices of $-\nabla f(x)$. Conn, Gould, and Toint [10] observe that $\chi(x)$ can be interpreted as the progress that can be made on a first-order model at x in a ball of radius unity with the constraint of preserving feasibility. They go on to observe that $\chi(x)$ is a direct generalization of $\| \nabla f(x) \|$; in fact, $\chi(x) = \| \nabla f(x) \|$ whenever $\Omega = \mathbb{R}^n$ or $x - \nabla f(x) \in \Omega$.

The work in [21, 22] used the measure $q(x)$ defined in (1.4) (this quantity appears in [10] as equation (12.1.19)), but the resulting stationarity result is unsatisfying in the case of general linear constraints. The quantity $\chi(x)$ turns out to be easier to work with than $q(x)$. The latter involves a projection onto the feasible polyhedron, and if the constraints binding at the projection do not correspond to the constraints near x, technical difficulties ensue in relating $q(x)$ to the geometry of the feasible region near x. This is not the case with $\chi(x)$.

Appendix B. Geometric results on cones and polyhedra. Here we present geometrical results having to do with our use of $\chi(\cdot)$ as a measure of stationarity. Since these are largely technical in nature, we have relegated them to an appendix.

The first proposition says that if one can move from x to $x+v$ and remain feasible, then v cannot be too outward-pointing with respect to the constraints near x. Recall from §2.1 that given a convex cone K and any vector v, there is a unique closest point of K to v, the *projection* of v onto K, which we denote by v_K. Thus $v_{N(x,\varepsilon)}$ is the projection of v onto the ε-normal cone $N(x,\varepsilon)$ while $v_{T(x,\varepsilon)}$ is the projection of v onto the ε-tangent cone $T(x,\varepsilon)$.

PROPOSITION B.1. *If $x \in \Omega$ and $x + v \in \Omega$, then for any $\varepsilon \geq 0$, $\| v_{N(x,\varepsilon)} \| \leq \varepsilon/\nu_{\min}$, where $\nu_{\min}$ is the constant from (2.4).*

Proof. Let $N = N(x,\varepsilon)$. The result is immediate if $v_N = 0$, so we need only consider the case when $v_N \neq 0$. Recall that N is generated by the outward-pointing normals to the binding constraints within distance ε of x; thus, the set $\mathcal{A} = \{ \, a_i \; | \; i \in I(x,\varepsilon) \, \}$ generates N. A simple calculation shows that the distance from x to $\{ \, y \; | \; a_i^T y = b_i \, \}$ is $(b_i - a_i^T x)/\| a_i \|$, so it follows that

$$\frac{b_i - a_i^T x}{\| a_i \|} \leq \varepsilon \quad \text{for all } i \in I(x,\varepsilon).$$

Meanwhile, since $x + v \in \Omega$, we have

$$a_i^T x + a_i^T v \leq b_i \quad \text{for all } i.$$

The preceding two relations then lead to

$$a_i^T v \leq b_i - a_i^T x \leq \varepsilon \, \| a_i \| \quad \text{for all } i \in I(x,\varepsilon).$$

Since N is generated by $\mathcal{A} \subseteq \mathbf{A} = \{a_1, \ldots, a_m\}$ and $v_N \neq 0$, by (2.1) and (2.4),

$$\nu_{\min} \, \| v_N \| \leq \max_{i \in I(x,\varepsilon)} \frac{v^T a_i}{\| a_i \|} \leq \max_{i \in I(x,\varepsilon)} \frac{\varepsilon \, \| a_i \|}{\| a_i \|} = \varepsilon. \qquad \square$$

For $x \in \Omega$ and $v \in \mathbb{R}^n$, define

$$\hat{\chi}(x;v) = \max_{\substack{x+w \in \Omega \\ \| w \| \leq 1}} w^T v. \tag{B.1}$$

Note from (A.1) that $\chi(x) = \hat{\chi}(x; -\nabla f(x))$. We use v in (B.1) to emphasize that the following results are purely geometric facts about cones and polyhedra.

The following proposition relates $\hat{\chi}(x;v)$ to the projection of v onto the cones $T(x,\varepsilon)$ and $N(x,\varepsilon)$. Roughly speaking, it says that if $\varepsilon > 0$ is small, so that we are only looking at a portion of the boundary very near x, then the projection of v onto $T(x,\varepsilon)$ (i.e., the portion of v pointing into the interior of the feasible region) cannot be small unless $\hat{\chi}(x;v)$ is also small.

PROPOSITION B.2. *If $x \in \Omega$, then for all $\varepsilon \geq 0$,*

$$\hat{\chi}(x;v) \leq \| v_{T(x,\varepsilon)} \| + \frac{\varepsilon}{\nu_{\min}} \, \| v_{N(x,\varepsilon)} \|,$$

where $\nu_{\min}$ is the constant from (2.4).

Proof. Let $N = N(x,\varepsilon)$ and $T = T(x,\varepsilon)$. Writing v in terms of its polar decomposition, $v = v_N + v_T$, we obtain

$$\hat{\chi}(x;v) = \max_{\substack{x+w \in \Omega \\ \| w \| \leq 1}} w^T v \leq \max_{\substack{x+w \in \Omega \\ \| w \| \leq 1}} w^T v_T + \max_{\substack{x+w \in \Omega \\ \| w \| \leq 1}} w^T v_N.$$

For the first term on the right-hand side we have

$$\max_{\substack{x+w\in\Omega \\ \|w\|\le 1}} w^T v_T \le \| v_T \|.$$

Meanwhile, for any w we have

$$w^T v_N = (w_T + w_N)^T v_N \le w_N^T v_N$$

since $w_T^T v_N \le 0$. Thus,

$$\max_{\substack{x+w\in\Omega \\ \|w\|\le 1}} w^T v_N \le \max_{\substack{x+w\in\Omega \\ \|w\|\le 1}} \| w_N \| \, \| v_N \|.$$

However, since $x + w \in \Omega$, Proposition B.1 tells us that

$$\| w_N \| \le \frac{\varepsilon}{\nu_{\min}}.$$

Therefore,

$$\hat{\chi}(x; v) \le \| v_T \| + \frac{\varepsilon}{\nu_{\min}} \| v_N \|. \qquad \square$$

REFERENCES

[1] L. Armijo, *Minimization of functions having Lipschitz continuous first partial derivatives*, Pacific Journal of Mathematics, 16 (1966), pp. 1–3.

[2] C. Audet and J. E. Dennis, Jr., *Analysis of generalized pattern searches*, SIAM Journal on Optimization, 13 (2003), pp. 889–903.

[3] D. M. Avis, *A C implementation of the reverse search vertex enumeration algorithm*, Computational Geometry and Discrete Geometry (Surikaisekikenkyusho-Kokyuroku), 872 (1994), pp. 16–29.

[4] ———, `lrs` *homepage*. `http://cgm.cs.mcgill.ca/~avis/C/lrs.html`, 2001. McGill University, Montreal.

[5] D. M. Avis and K. Fukuda, *A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra*, Discrete and Computational Geometry, 8 (1992), pp. 295–313.

[6] ———, *Reverse search for enumeration*, Discrete Applied Mathematics, 65 (1996).

[7] F. H. Clarke, *Optimization and Nonsmooth Analysis*, Wiley, New York, 1983. Reissued as Classics in Applied Mathematics, Vol. 5, SIAM Publications, Philadelphia, 1990.

[8] ———, *Optimization and Nonsmooth Analysis*, Society for Industrial and Applied Mathematics, Philadelphia, 1990.

[9] A. R. Conn, N. Gould, A. Sartenaer, and P. L. Toint, *Convergence properties of an augmented Lagrangian algorithm for optimization with a combination of general equality and linear constraints*, SIAM Journal on Optimization, 6 (1996), pp. 674–703.

[10] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust-Region Methods*, vol. 1 of MPS/SIAM Series on Optimization, SIAM, Philadelphia, 2000.

[11] E. D. Dolan, R. M. Lewis, and V. J. Torczon, *On the local convergence properties of pattern search*, SIAM Journal on Optimization, 14 (2003), pp. 567–583.

[12] K. Fukuda, `cdd`*,* `cddplus` *and* `cddlib` *homepage*. `http://www.cs.mcgill.ca/~fukuda/soft/cdd_home/cdd.html`, 2004. McGill University, Montreal.

[13] A. A. Goldstein, *Constructive Real Analysis*, Harper & Row, New York, 1967.

[14] L. Grippo, F. Lampariello, and S. Lucidi, *Global convergence and stabilization of unconstrained minimization methods without derivatives*, Journal of Optimization Theory and Applications, 56 (1988), pp. 385–406.

[15] R. Hooke and T. A. Jeeves, *Direct search solution of numerical and statistical problems*, Journal of the Association for Computing Machinery, 8 (1961), pp. 212–229.

[16] T. G. Kolda, *Revisiting asynchronous parallel pattern search for nonlinear optimization*, Tech. Rep. SAND2004-8055, Sandia National Laboratories, Livermore, California, February 2004.

[17] T. G. Kolda, R. M. Lewis, and V. Torczon, *Optimization by direct search: New perspectives on some classical and modern methods*, SIAM Review, 45 (2003), pp. 385–482.

[18] ———, *Convergence properties of an augmented Lagrangian direct search algorithm for optimization with a combination of general equality and linear constraints*. In preparation, 2004.

[19] R. M. Lewis, A. Shepherd, and V. Torczon, *Implementing generating set search methods for linearly constrained minimization*. In preparation, 2004.

[20] R. M. Lewis and V. Torczon, *Pattern search algorithms for bound constrained minimization*, SIAM Journal on Optimization, 9 (1999), pp. 1082–1099.

[21] ———, *Pattern search methods for linearly constrained minimization*, SIAM Journal on Optimization, 10 (2000), pp. 917–941.

[22] ———, *A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds*, SIAM Journal on Optimization, 12 (2002), pp. 1075–1089.

[23] S. Lucidi and M. Sciandrone, *Numerical results for unconstrained optimization without derivatives*, in Nonlinear Optimization and Applications, G. Di Pillo and F. Giannessi, eds., Kluwer Academic/Plenum Publishers, New York, 1996, pp. 261–270. Proceedings of the International School of Mathematics "G. Stampacchia" 21st Workshop, Erice, Italy, June 13–21, 1995.

[24] ———, *A derivative-free algorithm for bound constrained optimization*, Computational Optimization and Applications, 21 (2002), pp. 119–142.

[25] ———, *On the global convergence of derivative-free methods for unconstrained optimization*, SIAM Journal on Optimization, 13 (2002), pp. 97–116.

[26] S. Lucidi, M. Sciandrone, and P. Tseng, *Objective-derivative-free methods for constrained optimization*, Mathematical Programming, 92 (2002), pp. 37–59.

[27] J. H. May, *Linearly Constrained Nonlinear Programming: A Solution Method That Does Not Require Analytic Derivatives*, PhD thesis, Yale University, New Haven, Connecticut, December 1974.

[28] J. J. Moreau, *Décomposition orthgonale d'un espace hilbertien selon deux cônes mutuellement polaires*, Comptes Rendus de l'Académie des Sciences de Paris, 255 (1962), pp. 238–240.

[29] T. S. Motzkin, H. Raiffa, G. L. Thompson, and R. M. Thrall, *The double description method*, in Contributions to Theory of Games, H. W. Kuhn and A. W. Tucker, eds., vol. 2, Princeton University Press, 1953.

[30] J. Nocedal and S. Wright, *Numerical Optimization*, Springer Series in Operations Research, Springer, New York, 1999.

[31] C. J. Price and I. D. Coope, *Frames and grids in unconstrained and linearly constrained optimization: A nonsmooth approach*, SIAM Journal on Optimizaiton, 14 (2003), pp. 415–438.

[32] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[33] V. Torczon, *On the convergence of pattern search algorithms*, SIAM Journal on Optimization, 7 (1997), pp. 1–25.

[34] J. van Tiel, *Convex Analysis: An Introductory Text*, John Wiley & Sons, 1984.

[35] P. Wolfe, *Convergence conditions for ascent methods*, SIAM Review, 11 (1969), pp. 226–235.

[36] W. Yu and Y. Li, *A direct search method by the local positive basis for linearly constrained optimization*, Chinese Annals of Mathematics, 2 (1981), pp. 139–145.

[37] E. H. Zarantonello, *Projections on convex sets in Hilbert space and spectral theory*, in Contributions to Nonlinear Functional Analysis, E. H. Zarantonello, ed., Academic Press, London and New York, 1971, pp. 237–424.